System wejścia-wyjścia

System wejścia-wyjścia

Trzy rodzaje urządzeń wejścia-wyjścia:

- Urządzenia pamięci (dyski, taśmy)
- Urządzenia przesyłania danych (karty sieciowe, modemy)
- Urządzenia komunikacji z człowiekiem (klawiatury, myszy, monitory)

Różnice między urządzeniami we-wy

- **Urządzenie znakowe** - przesyła bajty (znaki) z osobna jeden za drugim (terminal)
  Urządzenie blokowe - przesyła jednorazowo całe bloki (dysk)
- **Dostęp sekwencyjny** - dane przesyłane kolejno w sposób uporządkowany (modem)
  Dostęp swobodny - można mieć dostęp do danych w różnych miejscach, niekoniecznie kolejno (CD-ROM)
- **Przesyłanie synchroniczne** - taktowane zegarem (taśma)
  przesyłanie asynchroniczne - w nieokreślonych chwilach czasu, sterowane startem i stopem (klawiatura)
- **Urządzenie dzielone** - przez kilka procesów (dysk)
  Wyłączne - tylko dla jednego użytkownika (taśma)
- **Szybkość działania** - od B/s do GB/s
- **Kierunek przesyłania** - czytanie, pisanie lub czytanie i pisanie

Sterowanie urządzeniami wejścia-wyjścia

Przekazywanie poleceń z procesora do sterownika:

Sterownik posiada rejestry do pamiętania danych i sygnałów sterujących. Procesor pisze i czyta do rejestrów w sterowniku.

- Procesor posiada specjalne rozkazy do pisania i czytania portów,
- Operacje we-wy odbywają się w pamięci - rejestry są odwzorowywane w przestrzeni adresowej procesora.

W komputerach IBM PC zastosowane są obydwie metody:

- Dla kontrolera grafiki - ekran odwzorowany w pamięci
- Dla portów szeregowych - rejestry we-wy i bufory

Port wejścia-wyjścia

- Stan (czytane przez procesor: zakończenie wykonywania polecenia, dostępność bajtu do czytania, błąd urządzenia)
- Sterowanie (zapisywane przez procesor: rozpoczęcie polecenia, zmiana trybu pracy urządzenia)
- Dane wejściowe (czytane przez procesor dane właściwe)
- Dane wyjściowe (zapisywane przez procesor dane dla urządzenia)

Układ FIFO - „magazyn pośredni” - buforuje dane, których komputer lub urządzenie nie może w danej chwili odebrać.

Odpytywanie

Do uzgadniania pomiędzy procesorem a urządzeniem w prostym schemacie producent-konsument wystarczą dwa bity:

- od strony procesora **bit gotowości polecenia** w rejestrze pleceń - sygnalizujący kompletne polecenie dla urządzenia
- od strony urządzenia **bit zajętości** (w rejestrze stanu), sygnalizujący że urządzenie jest zajęte pracą.

Kolejność działań przy uzgadnianiu:

- Procesor realizuje aktywne czekanie, dopóki bit zajętości jest ustawiony
- Procesor ustawia bit pisania i wpisuje bajt danych do rejestru danych wy.
- Procesor ustawia bit gotowości polecenia
- Sterownik ustawia bit zajętości po zauważeniu bitu gotowości polecenia
- Sterownik czyta rejestr poleceń, rozpoznaje polecenie pisania. Czyta bajt danych z rejestru i wykonuje na urządzeniu operację wejścia-wyjścia
- Sterownik czyści bit gotowości polecenia, bit błedu, a na końcu bit zajętości

I tak dla każdego bajtu danych

Przerwania

Jeśli urządzenie jest rzadko gotowe do działania, odpytywanie staje się nieefektywne (procesor większość czasu poświęca na aktywne czekanie.

Mechanizm przerwań:

- procesor ma końcówkę (nóżkę) badającą stan linii zgłaszania przerwań po wykonaniu każdego rozkazu.
- Jeśli procesor wykryje wystąpienie przerwania, to wykonuje operacje zachowania stanu bieżącego procesu i przechodzi do procedur obsługi przerwań.
- Po wykonaniu niezbędnych operacji procesor wraca do wykonywania przerwanego zadania
- W nowoczesnych architekturach komputerów możliwa zaawansowana obsługa przerwań:
  - opóźnianie obsługi przerwania podczas działań krytycznych,
  - maskowanie przerwań (dwie linie przerwań - maskowalna i niemaskowalna)
  - przerwania wielopoziomowe o różnym priorytecie

Bezpośredni dostęp do pamięci

- Jest używany w celu uniknięcia transmisji bajt-po-bajcie (zwanego programowanym wejściem-wyjściem) dla urządzeń transmitujących wielkie ilości danych (np dysk), co oszczędza wiele cykli procesora
- Wiele procedur związanych z transmisją jest wtedy wykonywana przez specjalizowany procesor - **sterownik bezpośredniego dostępu do pamięci (DMA controller)**
- Przed rozpoczęciem transmisji w trybie DMA, procesor zapisuje w pamięci blok sterujący DMA (wskaźnik do źródła, adres docelowy, liczba bajtów do przesłania), następnie przesyła do sterownika DMA adres tego bloku i przechodzi do wykonywania innych prac.
- Sterownik DMA wykonuje transmisję, przejmując w tym czasie sterownie szyną pamięci. Procesor nie ma wtedy dostępu do pamięci, ale może korzystać z cache i rejestrów.

Wejście-wyjście z blokowaniem

- Blokowanie uwalnia procesor od aktywnego czekania - proces przenoszony jest do kolejki procesów czekających. Po zakończeniu we-wy proces przechodzi do kolejki procesów gotowych.
- Niektóre procesy wymagają wejścia-wyjścia bez blokowania, np proces w którym sygnały z klawiatury lub myszy przeplatają się z przetwarzaniem i wyświetlaniem na ekranie, albo czytanie z dysku z dekompresją danych.
- W aplikacjach wielowątkowych można zablokować pewne wątki, a inne zostawić aktywne.

Podsystem wejścia-wyjścia w jądrze

- **Planowanie wejścia-wyjścia** ma na celu poprawę wydajności systemu, polepszenie wspólnego korzystania z urządzeń przez procesy i zmniejszenie średniego czasu oczekiwania.
- **Buforowanie** - dopasowanie prędkości „producenta” i „konsumenta” danych (podwójne buforowanie), dopasowanie urządzeń operujących na różnych wielkościach bloków danych (np pakiety sieci a bloki na dysku)
- **Przechowywanie podręczne** (caching) - zapamiętanie kopii danych w szybkiej pamięci podręcznej
- **Spooling** - użycie bufora do przechowywania danych przeznaczonych dla urządzenia, które nie dopuszcza przeplatania danych z różnych procesów (np drukarka)
- **Obsługa błędów**
- **Struktury danych jądra** - jądro musi przechowywać informacje o stanie używanych składowych wejścia-wyjścia

Podsystem wejścia-wyjścia w jądrze, cd

Podsystem nadzoruje:

- zarządzanie przestrzenią nazw plików i urządzeń,
- przebieg dostępu do plików i urządzeń,
- poprawność formalną operacji,
- przydzielanie miejsca w systemie plików,
- przydział urządzeń,
- buforowanie, caching oraz spooling,
- planowanie operacji wejścia-wyjścia
- doglądanie stanu urządzeń, obsługę błędów, czynności naprawcze po awarii.
- konfigurowanie i wprowadzanie w stan początkowy modułu sterującego

Poprawianie wydajności wejścia-wyjścia

Podsystem ma na celu:

- zmniejszać liczbę przełączeń kontekstu,
- zmniejszać liczbę kopiowań danych w pamięci podczas przekazywania od urządzenia do aplikacji,
- zmniejszać częstość przerwań poprzez przesyłanie dużych porcji informacji, przez optymalizację sterowników i stosowanie odpytywania, gdzie to korzystne,
- zwiększać współbieżność poprzez stosowanie sterowników pracujących w trybie DMA,
- realizować elementarne działania za pomocą sprzętu i pozwalać na ich współbieżne wykonywanie w sterownikach,
- równoważyć wydajność procesora, podsystemów pamięci, szyny i operacji wejścia-wyjścia

Elementy od których zależy wydajność dysku

Czytanie lub pisanie wymaga ustawienia głowicy nad określoną ścieżką i na początku konkretnego sektora. Potrzebny na to jest **czas dostępu -** suma czasu przeszukiwania i opóźnienia obrotowego

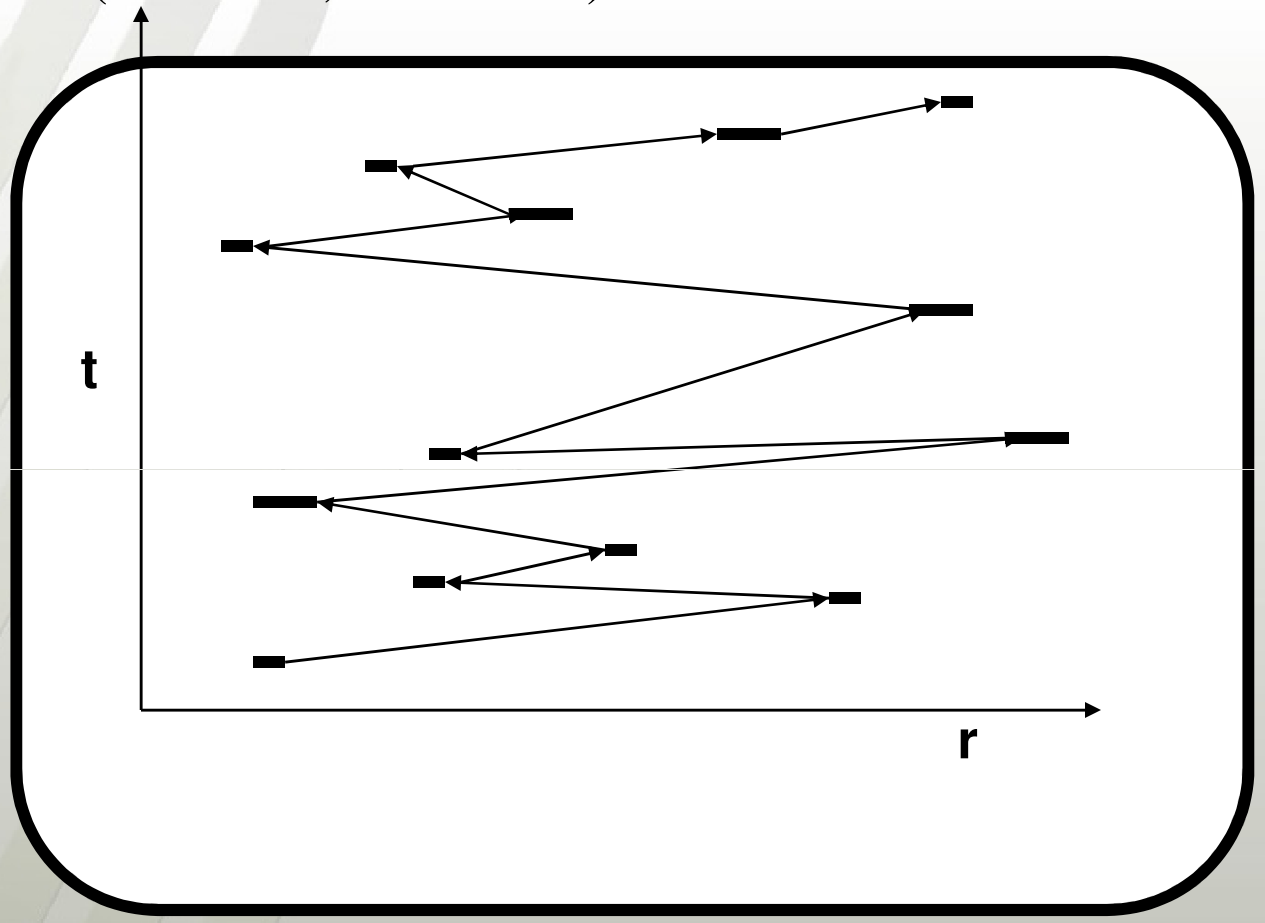
- czas przeszukiwania (seek time) - czas potrzebny na ustawienie głowicy nad ścieżką
- opóźnienie obrotowe (rotational latency) - czas potrzebny na obrót właściwego sektora pod głowicę,

Czas przeszukiwania ma duży wpływ na wydajność.

Możemy go optymalizować, ponieważ system operacyjny zarządza kolejką żądań dysku

Planowanie wejścia-wyjścia dla dysku

- FCFS - (first come, first served)



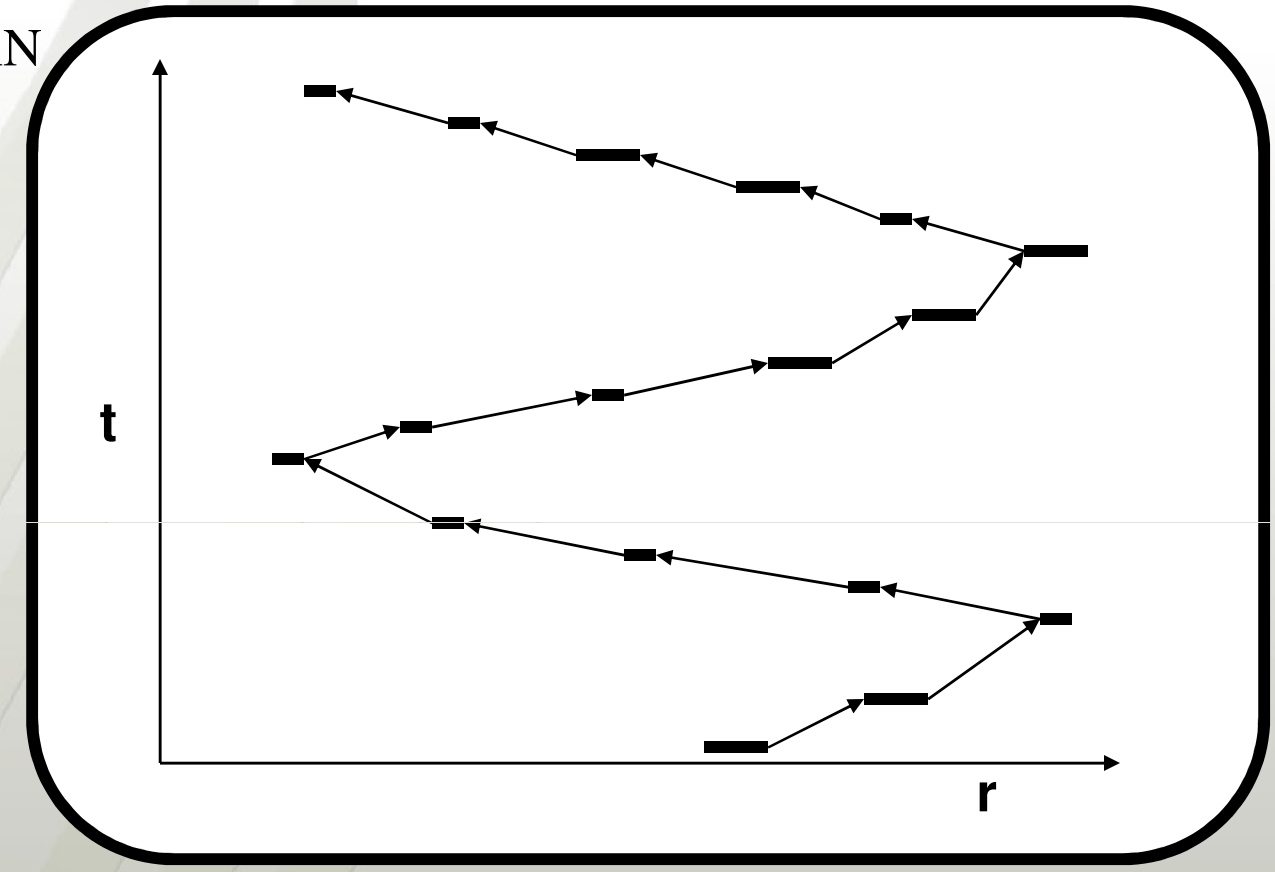
żądania wejścia-wyjścia realizowane w kolejności zgłaszania. Algorytm sprawiedliwy, ale mało wydajny. Przy dużej liczbie zgłoszeń planowanie zbliżone do losowego (najgorszego)

Planowanie wejścia-wyjścia dla dysku, cd

- PRI (*priority*) - krótkie zadania wsadowe i interakcyjne mają wyższy priorytet i pierwszeństwo w dostępie do dysku. Rozwiązanie nieefektywne dla baz danych.
- LIFO (*last in-first out*) - minimalizuje ruch głowicy (żądania korzystają z ograniczonego obszaru na dysku). Możliwość głodzenia procesów
- SSTF (*shortest seek time first*) - najmniejszy ruch głowicy od pozycji bieżącej. Zawsze minimalny czas przeszukiwania. Możliwe głodzenie.

Planowanie wejścia-wyjścia dla dysku, cd

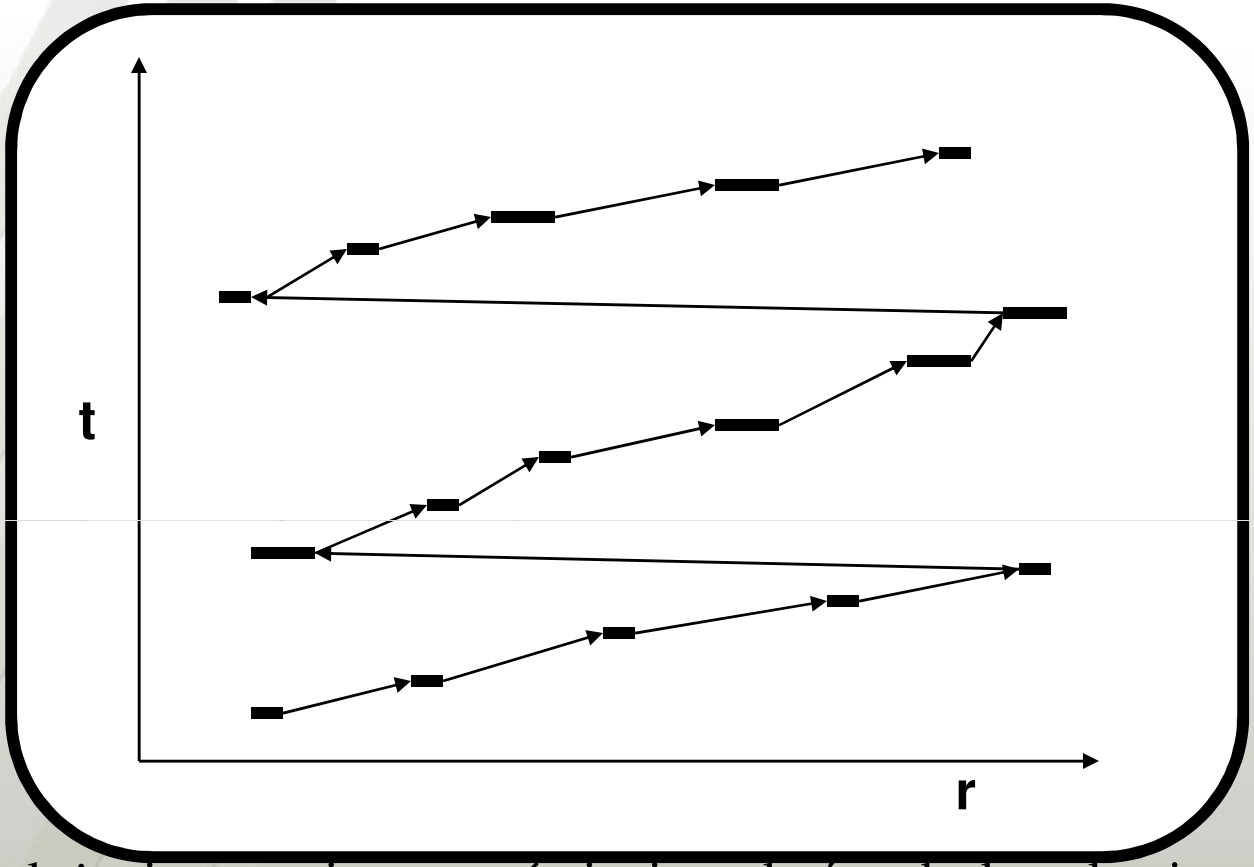
- SCAN



głowica przesuwa się od jednej krawędzi dysku do drugiej, lub do ostatniego żądania. Potem zmienia się kierunek przesuwania. Działa jak winda - najpierw zamówienia „w górę”, potem „w dół”. Najszybciej po sobie obsługiwane żądania na skrajnych ścieżkach, a najpóźniej - po przeciwnej stronie dysku. Nie ma głodzenia

Planowanie wejścia-wyjścia dla dysku, cd

- C-SCAN (circular scan)



po obsłużeniu ostatniego zamówienia na końcu dysku, głowica wraca na początek i zaczyna od zera.

- N-step-SCAN - podział kolejki na podkolejki o długości N. Kolejne podkolejki obsługiwane metodą SCAN do końca.
- FSCAN - dwie kolejki, jedna realizowana, nowe żądania przychodzą do drugiej

Użytkowy interfejs we-wy drivery urządzeń

- Moduły sterujące - wewnętrznie dostosowane są do konkretnych urządzeń, a zewnętrznie udostępniają pewien standardowy interfejs
- standaryzacja pomaga producentom sprzętu tworzyć sterowniki do własnych urządzeń, widziane przez „obce” systemy operacyjne
- Systemy muszą umożliwiać instalację sterowników do nowego sprzętu

Zegary i czasomierze

Spełniają trzy podstawowe funkcje:

- podawanie bieżącego czasu,
- podawanie upływającego czasu,
- powodowanie wykonania określonej operacji w określonej chwili

c**zasomierz programowalny** - służy do pomiaru upływającego czasu i powodowania wykonania operacji w zadanym czasie

- można go zaprogramować na określony czas, po którym generuje on przerwanie
- jest to też zegar systemowy do taktowania kwantów czasu (dla przydziału procesora)

Urządzenia sieciowe

- Sieciowe wejście-wyjście różni się znacznie od dyskowego, pod względem wydajności i adresowania
- Interfejs gniazda (socket) - aplikacje umożliwiają tworzenie gniazda, połączenie lokalnego gniazda ze zdalnym adresem, nasłuchiwanie, przesyłanie i odbieranie pakietów za pomocą połączenia
- funkcja **wybierz** zarządza gniazdami
- W systemach Windows NT/2000 - interfejs do kontaktowania się z kartą sieciową oraz interfejs do do protokołów sieciowych
- W systemie Unix - półdupleksowe potoki, pełnodupleksowe kolejki FIFO, pełnodupleksowe strumienie, kolejki komunikatów i gniazda

Wejście-wyjście w różnych systemach operacyjnych

MS-DOS

- Nazwy urządzeń zakończone dwukropkiem, np. A:, C:, PRN:
- Odwzorowane są na określone adresy portów za pomocą tablicy urządzeń.
- System operacyjny może przypisać każdemu urządzeniu dodatkowych funkcji, np spooling dla drukarki

UNIX

- Nazwy urządzeń są pamiętane w przestrzeni nazw systemu plików (katalog /dev /devices)
- W strukturze katalogowej zamiast numeru I-węzła jest zapisany numer urządzenia w postaci pary <starszy,młodszy>.
- **Starszy** numer urządzenia identyfikuje moduł sterujący, który trzeba wywołać dla obsługi we-wy, a **młodszy** jest przekazywany do modułu sterującego jako indeks do tablicy urządzeń.
- Wpis w tablicy urządzeń zawiera adres portu lub adres sterownika odwzorowany w pamięci

Wejście-wyjście w różnych systemach operacyjnych

UNIX -cd

- **Strumień** - kanał komunikacyjny pomiędzy procesem użytkownika a urządzeniem
- Strumień składa się z:
  - głowy - interfejs z programem użytkownika,
  - zakończenia sterującego - interfejs z urządzeniem
  - modułów przetwarzających - filtrów na komunikatach płynących w strumieniu.
- Każdy ze składników strumienia zawiera co najmniej jedną parę kolejek: we i wy
- Kontrola przepływu w strumieniu realizowana jest za pomocą buforów wejścia-wyjścia
- We-wy na strumieniu jest asynchroniczne
- Sterownik jest zawieszony do chwili pojawienia się danych
- Gdy bufor jest pełny - utrata komunikatów

Wejście-wyjście w różnych systemach operacyjnych

Windows NT/2000

- **Cache manager** - zarządza całym podsystemem wejścia-wyjścia
- Podsystem we-wy traktuje **File System Drivers** tak samo jak **Device Drivers**
- **Network Drivers** - sterowniki sieci
- **Hardware Device Drivers** - dostęp do rejestrów urządzeń wejścia-wyjścia
- Wejście-wyjście może być **asynchroniczne** i **synchroniczne**
- System ma wbudowane mechanizmy do sygnalizowania zakończenia asynchronicznego wejścia-wyjścia